SICHER IN DIE CLOUD MIT ANGULAR UND SPRING BOOT

KARLSRUHER ENTWICKLERTAG

22. MAI 2017

ANDREAS FALK

NOVATEC CONSULTING GMBH

andreas.falk@novatec-gmbh.de

@NT_AQE, @andifalk

AGILE TESTING DAYS

jax

25 YEARS OOP 2016 YOUR SOURCE OF TRUST
Conference 01 – 05 February 2016
Exhibition 02 – 04 February 2016

JAVA Mag

Javamagazin

jax

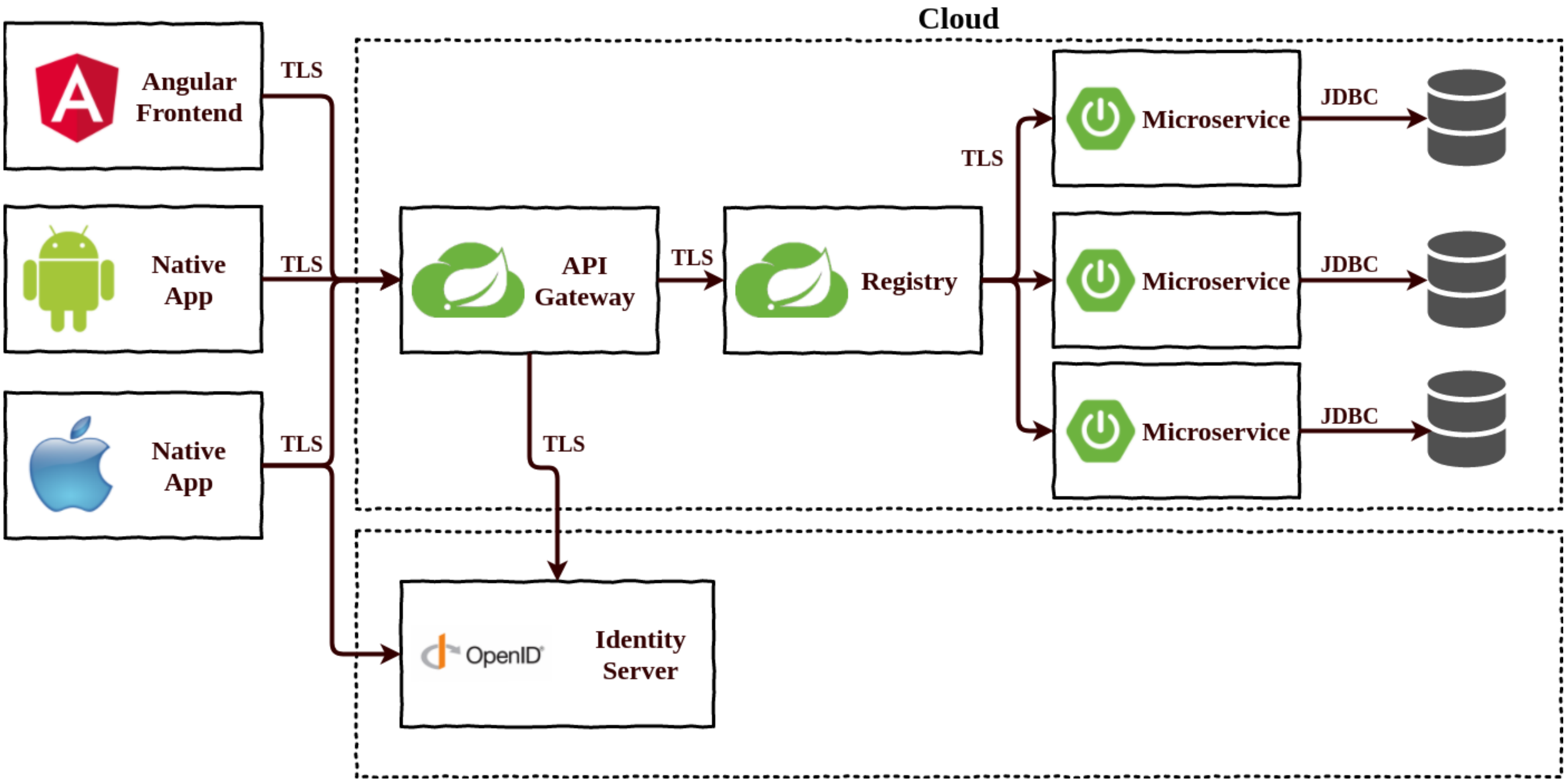
Der Lenz ist da

Spring 5

FRANKFURTER ENTWICKLERTAG

APPSEC EUROPE
ROMA MMXVI

ARCHITECTURE / THREAT MODEL


Cloud
Angular Frontend
Native App
Native App
TLS
TLS
TLS
API Gateway
TLS
Registry
TLS
Microservice
Microservice
Microservice
JDBC
JDBC
JDBC
TLS
OpenID
Identity Server

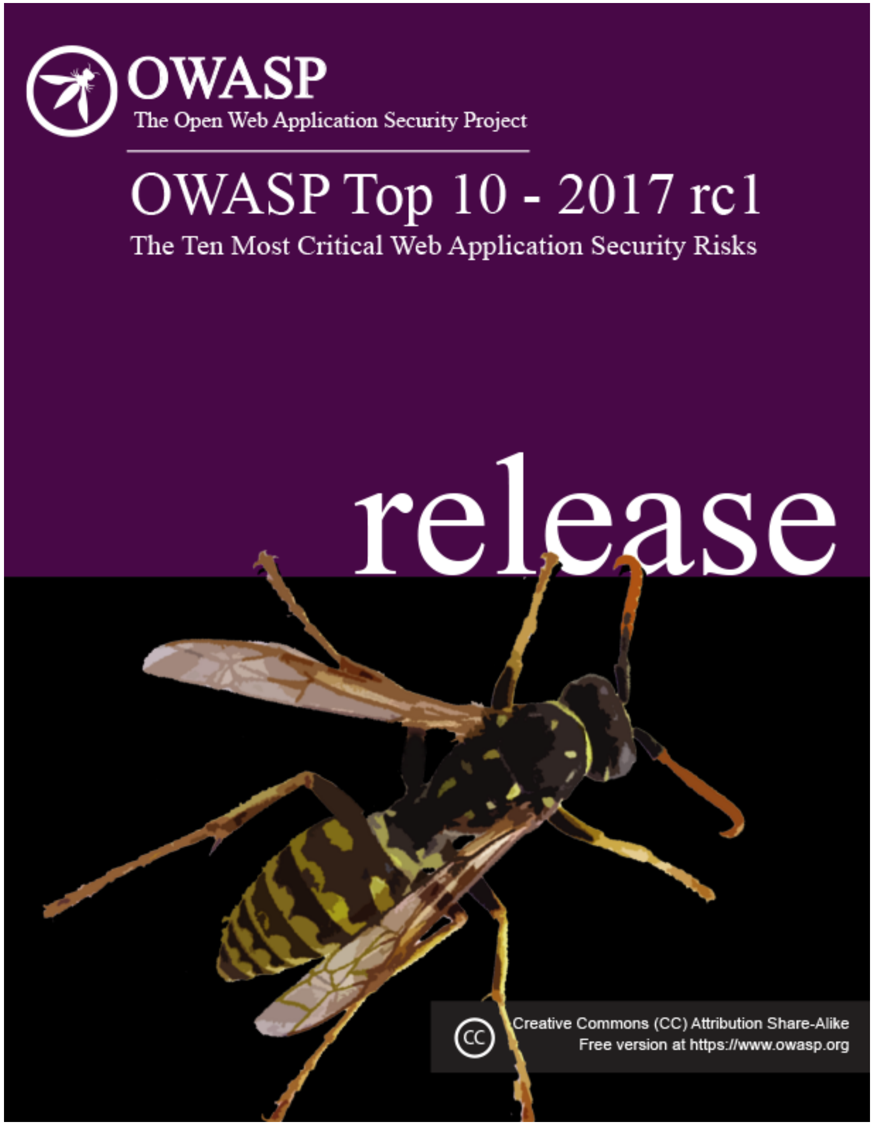
SQLInjection CSRF XSS OWASP OAuth2 OpenID-Connect AbUser-Stories Authentication

Authorization Secure Coding Security-Testing SSO DoS Sensitive-Data

Data-Privacy Crypto Code-Reviews Threat-Modeling Architecture Dependencies DAST

SAML SAST DevSecOps

SQLInjection CSRF XSS OAuth2 OpenID-Connect Authentication Authorization Secure Coding Security-Testing Sensitive-Data



OWASP

The Open Web Application Security Project

OWASP Top 10 - 2017 rc1

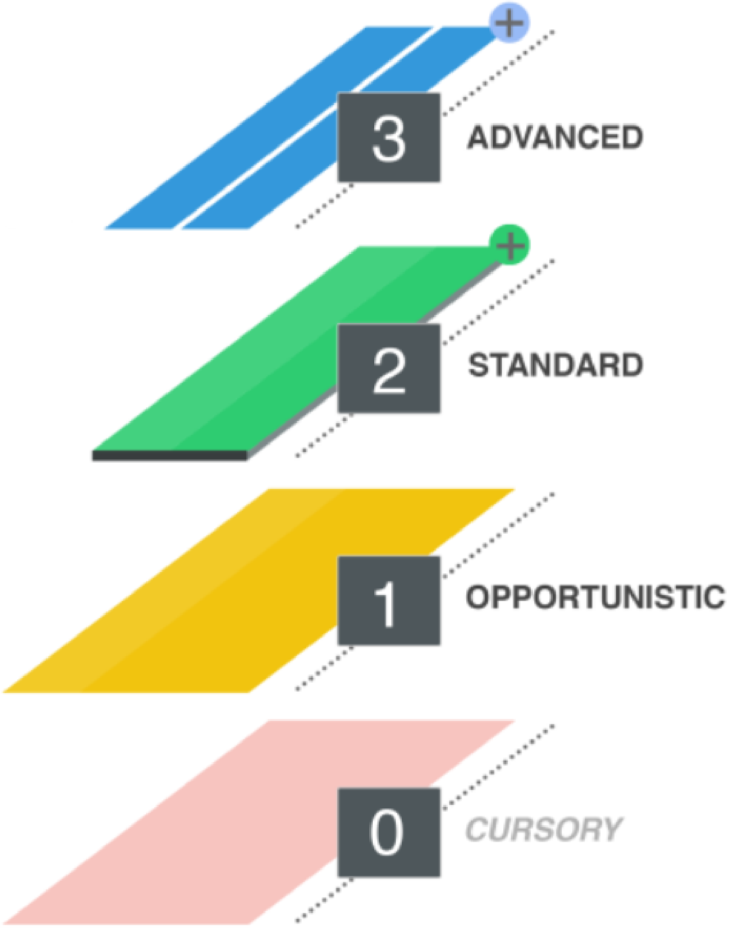
The Ten Most Critical Web Application Security Risks

release

Creative Commons (CC) Attribution Share-Alike
Free version at https://www.owasp.org

HTTPS://GITHUB.COM/OWASP/TOP10

APP SECURITY VERIFICATION STANDARD



https://github.com/OWASP/ASVS

PRO ACTIVE CONTROLS

OWASP ProActive CONTROLS

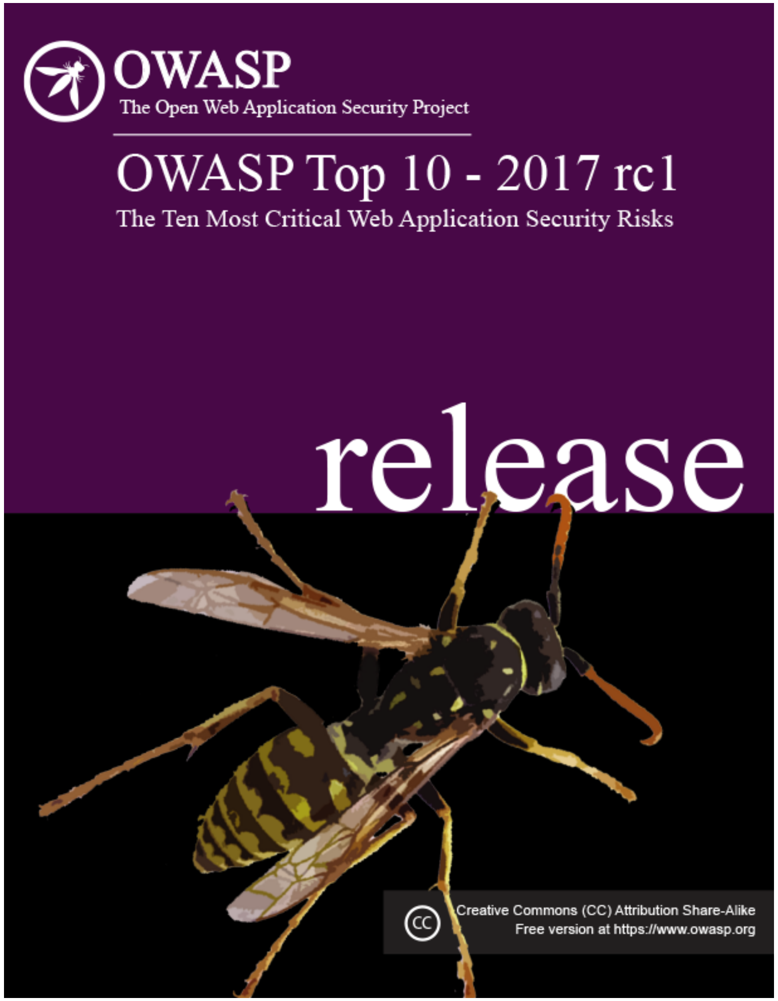
https://www.owasp.org/index.php/OWASP_Proactive_Controls

ANGULAR

ANGULARJS = ANGULAR 1

ANGULAR = ANGULAR 2.X, 4.X, ...

A3: CROSS-SITE SCRIPTING (XSS)



OWASP

The Open Web Application Security Project

OWASP Top 10 - 2017 rc1

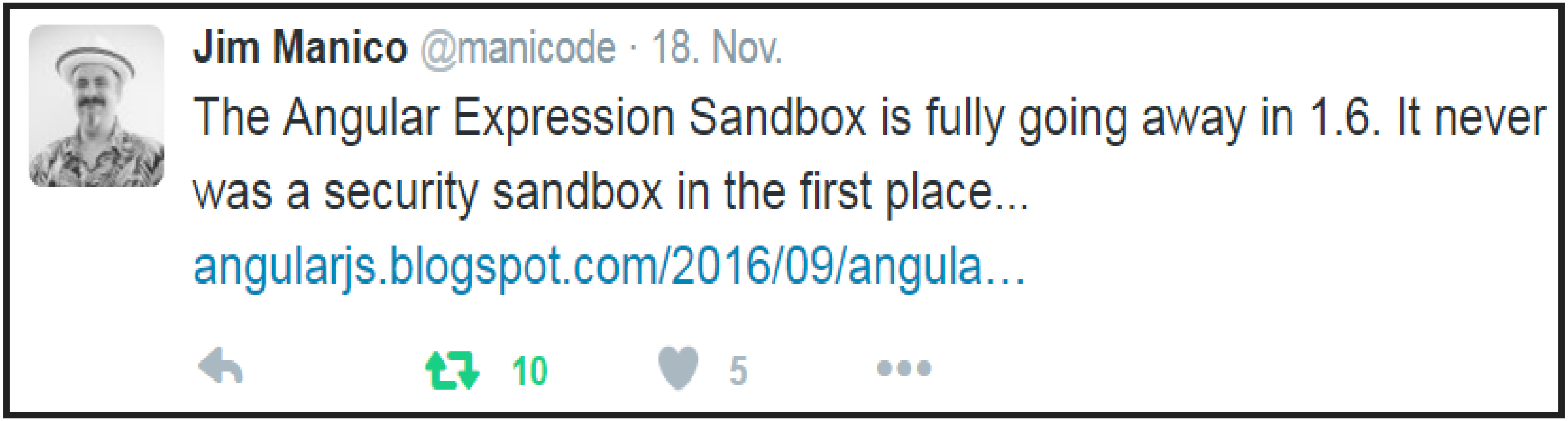
The Ten Most Critical Web Application Security Risks

release

Creative Commons (CC) Attribution Share-Alike

Free version at https://www.owasp.org

ANGULAR JS SECURITY



Jim Manico @manicode · 18. Nov.

The Angular Expression Sandbox is fully going away in 1.6. It never was a security sandbox in the first place...

angularjs.blogspot.com/2016/09/angula…

https://angularjs.blogspot.de/2016/09/angular-16-expression-sandbox-removal.html

ANGULAR SECURITY

“...The basic idea is to implement automatic, secure escaping for all values that can reach the DOM... By default, with no specific action for developers, Angular apps must be secure...”

https://github.com/angular/angular/issues/8511

ANGULAR XSS PROTECTION

ANGULAR TEMPLATE = SAFE

INPUT VALUES = UNSAFE

ANGULAR COMPONENT

TYPESCRIPT

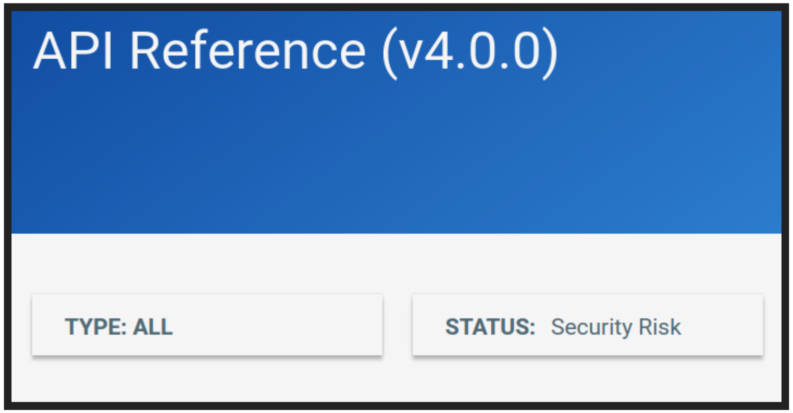
```
@Component({
    selector: 'app-root',
    templateUrl: 'app.component.html',
    styleUrls: ['app.component.css']
})
export class AppComponent {

    untrustedHtml:string =
        '<em><script>alert("hello")</script></em>';

}
```

ANGULAR TEMPLATE

HTML BINDINGS

```
<h2>Binding of potentially dangerous HTML-snippets</h2>

<h3>Encoded HTML snippet</h3>
<h3 class="trusted">{{untrustedHtml}}</h3>

<h3>Sanitized HTML snippet</h3>
<h3 class="trusted" [innerhtml]="untrustedHtml"></h3>
```

UNSAFE ANGULAR API'S


API Reference (v4.0.0)

TYPE: ALL

STATUS: Security Risk

ElementRef: Direct access to DOM!

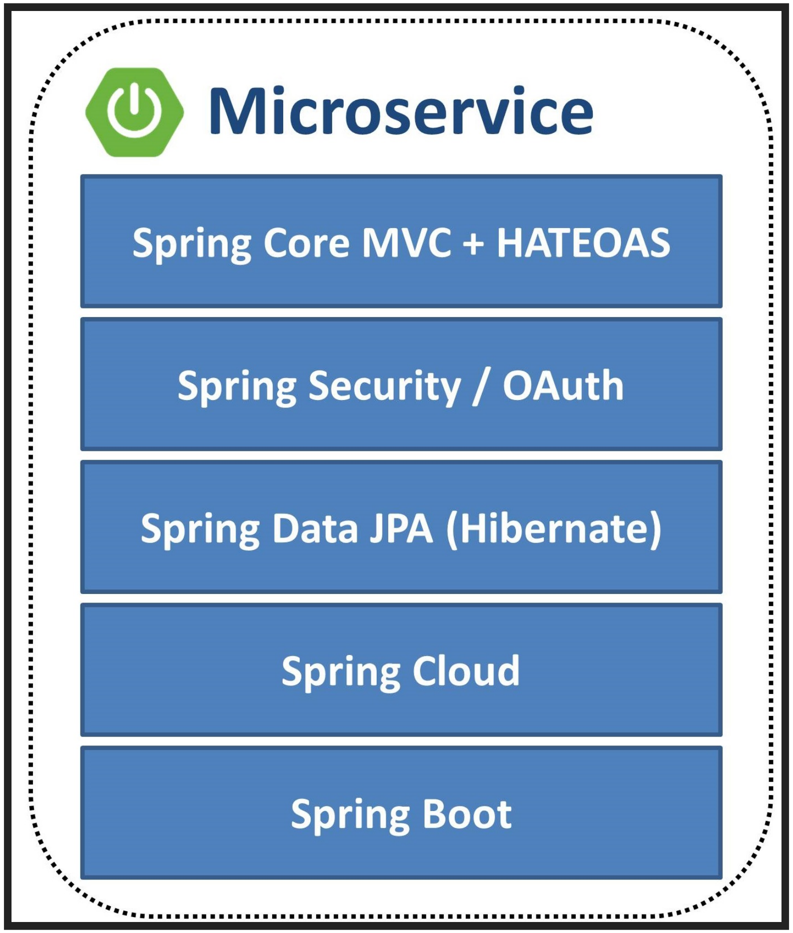
DomSanitizer: Deactivates XSS-Protection!

Do NOT use!

https://angular.io/docs/ts/latest

DEMO

BACKEND



Microservice

Spring Core MVC + HATEOAS

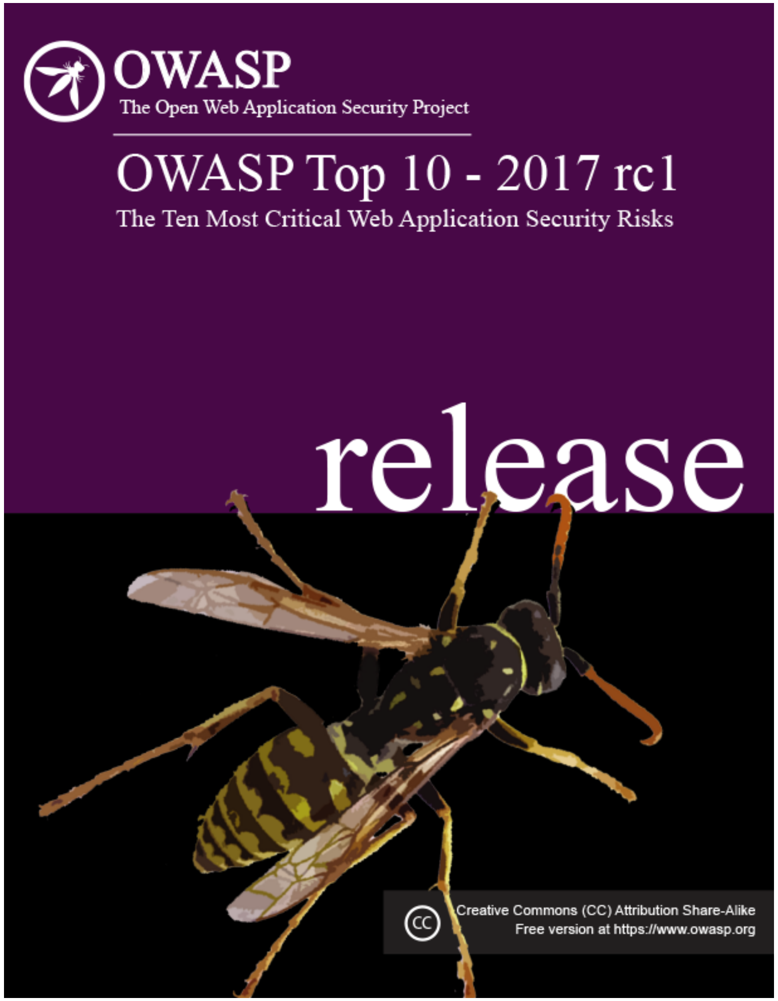
Spring Security / OAuth

Spring Data JPA (Hibernate)

Spring Cloud

Spring Boot

A1: INJECTION



OWASP

The Open Web Application Security Project

OWASP Top 10 - 2017 rc1

The Ten Most Critical Web Application Security Risks

release

Creative Commons (CC) Attribution Share-Alike

Free version at https://www.owasp.org

PERSISTENT XSS + INJECTIONS

STRONG TYPING + BEAN VALIDATION

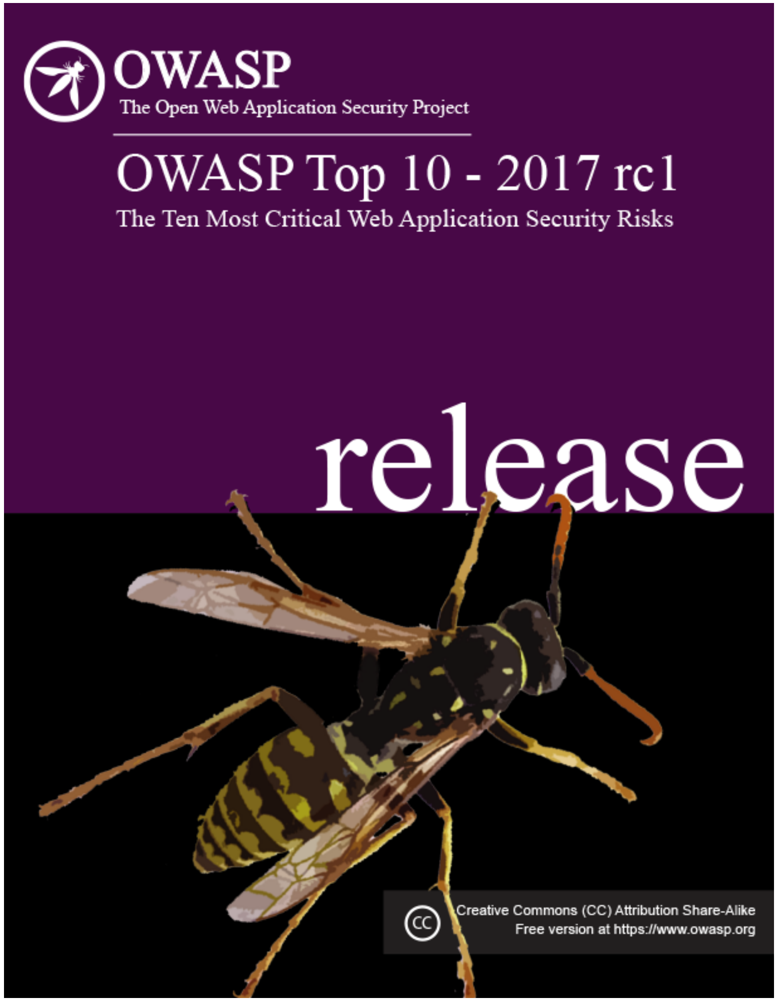
```
@Entity
public class Person extends AbstractPersistable<Long> {

   @NotNull
   @Pattern(regexp = "^[A-Za-z0-9- ]{1,30}$")
   private String lastName;

   @NotNull
   @Enumerated(EnumType.STRING)
   private GenderEnum gender;
   ...
}
```

SQL INJECTIONS

SPRING DATA JPA: USE PREPARED STATEMENTS

```
@Query(
"select u from User u where u.username = "
+ " :username and u.password = :password")
User findByUsernameAndPassword(
     @Param("username") String username,
     @Param("password") String password);
```

A8: CROSS-SITE REQUEST FORGERY (CSRF)



OWASP

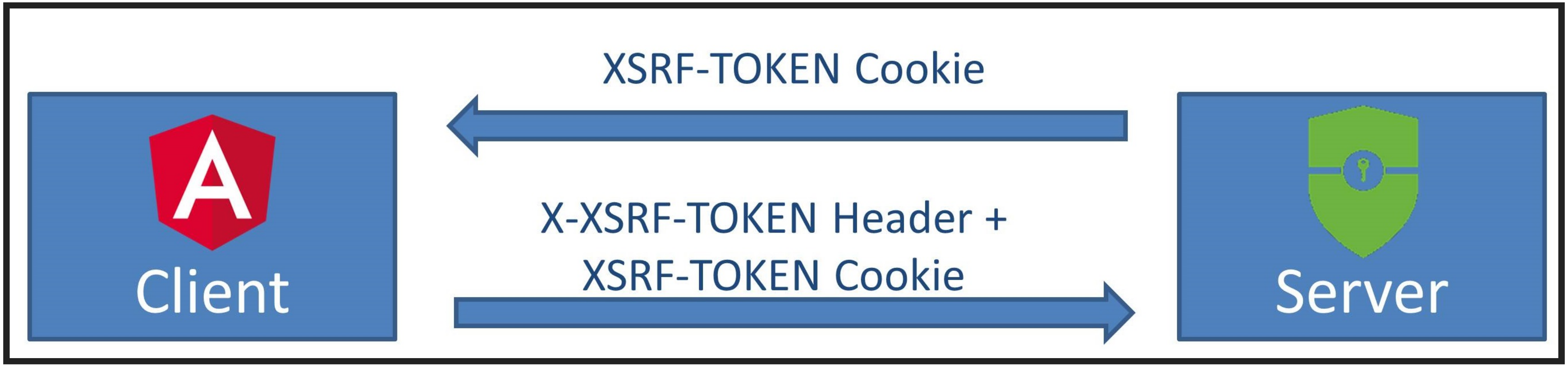
The Open Web Application Security Project

OWASP Top 10 - 2017 rc1

The Ten Most Critical Web Application Security Risks

release

DOUBLE SUBMIT CSRF TOKEN



SPRING SECURITY

SECURE BY DEFAULT

Authentication required for all HTTP endpoints

Session Fixation Protection

Session Cookie (HttpOnly, Secure)

CSRF Protection

Security Response Header

SPRING SECURITY CSRF CONFIGURATION

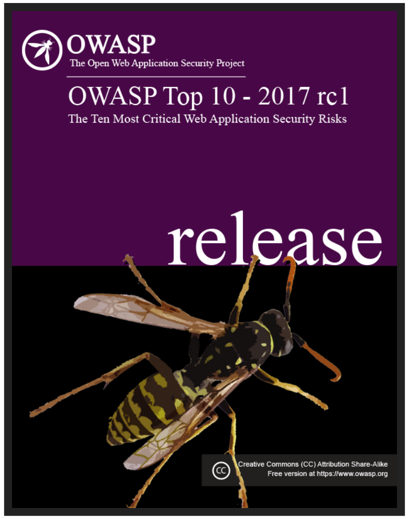
ANGULAR SUPPORT

```
@Configuration
public class WebSecurityConfiguration
    extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure(HttpSecurity http)
        throws Exception {
        …
        http
           .csrf().csrfTokenRepository(
              CookieCsrfTokenRepository.withHttpOnlyFalse()
           );
}
```

WHO AM I?

A2: BROKEN AUTHENTICATION AND SESSION MANAGEMENT

A10: UNDERPROTECTED APIS



OWASP
The Open Web Application Security Project

OWASP Top 10 - 2017 rc1
The Ten Most Critical Web Application Security Risks

release

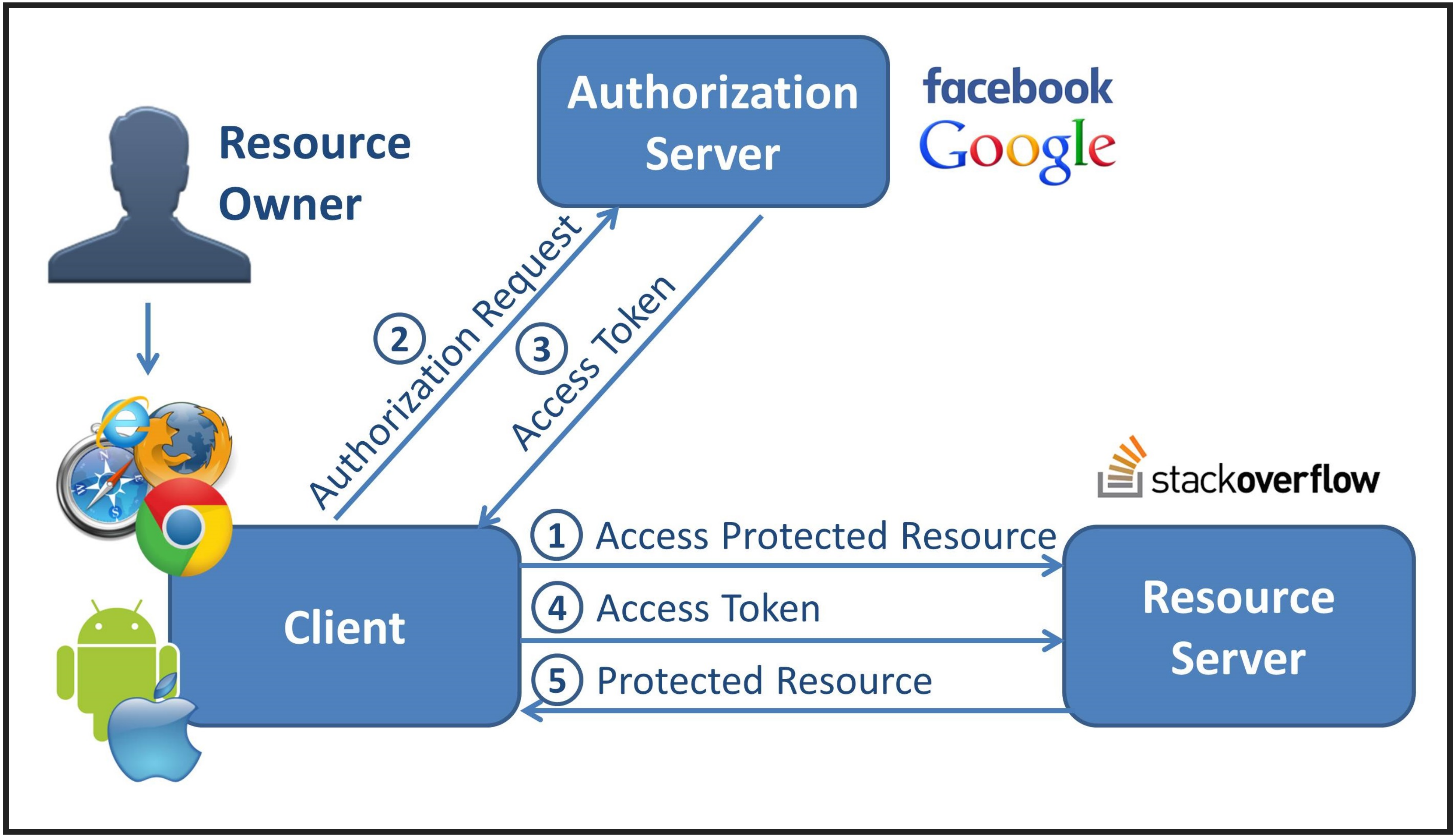
Creative Commons (CC) Attribution Share-Alike
Free version at https://www.owasp.org

AUTHENTICATION

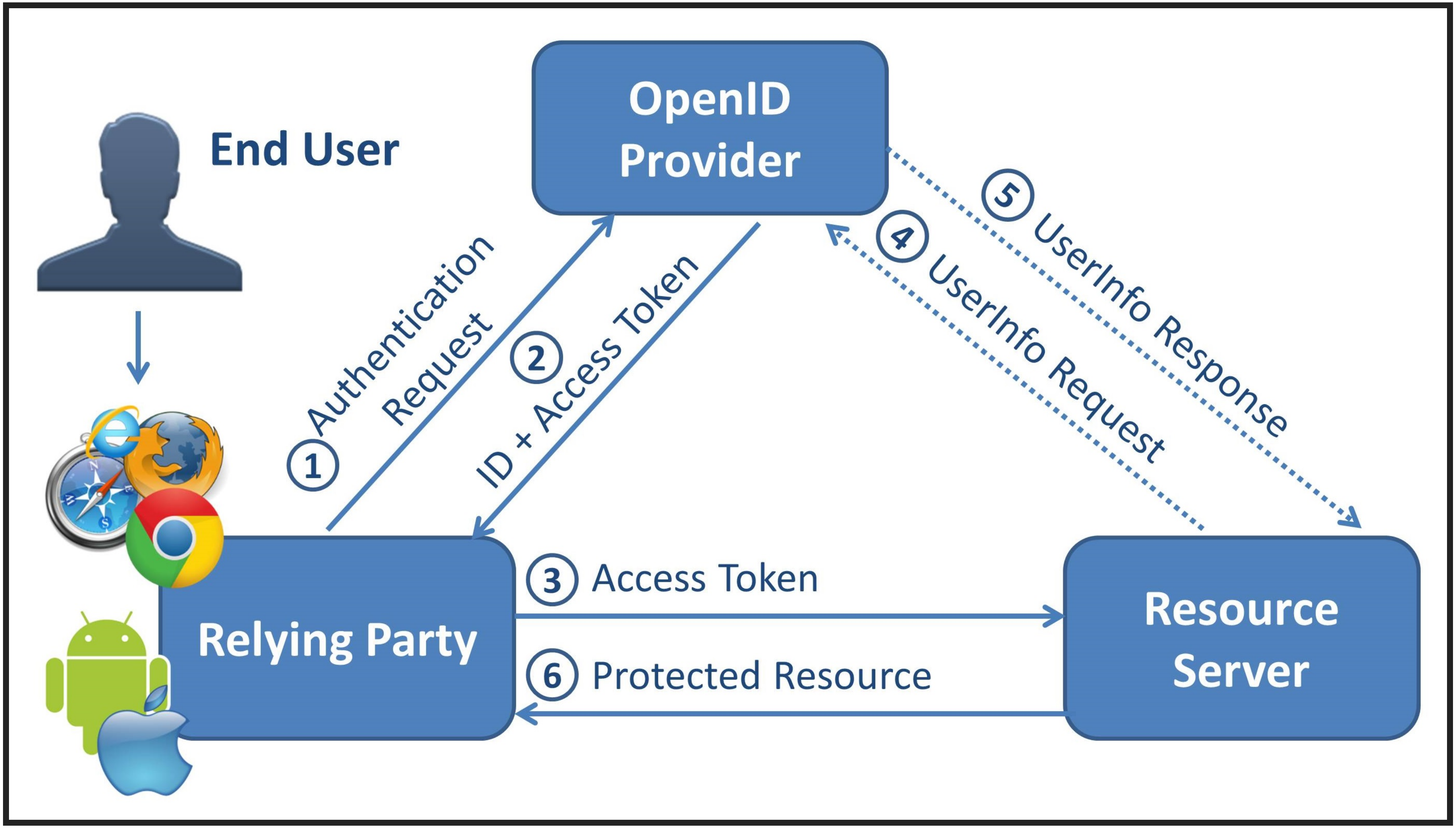
STATEFUL OR STATELESS?

Session Cookie	**Token (Bearer, JWT)**
With each Request (on same domain)	Manually as Header
Potential CSRF!	No CSRF possible
One domain	Cross domain (CORS)
Sensitive Info (HTTPS)	Sensitive Info (HTTPS)

OAUTH 2 = AUTHORIZATION



OPENID CONNECT = AUTHENTICATON



OAUTH 2 / OPENID CONNECT RESOURCE

```
@EnableResourceServer
@Configuration
public class OAuth2Configuration {
    @Bean
    public JwtAccessTokenConverterConfigurer
                jwtAccessTokenConverterConfigurer() {
        return new MyJwtConfigurer(...);
    }
    static class MyJwtConfigurer
            implements JwtAccessTokenConverterConfigurer {
        @Override
        public void configure(
            JwtAccessTokenConverter converter) {...}
    }
}
```

OAuth 2.0 Threat Model and Security Considerations

IMPLICIT GRANT

Validate...

...issuer identifier

...audiance (client id)

...signature (public key)

...expiration time

Implicit Client Implementer’s Guide
OAuth 2.0 Threat Model and Security Considerations

CLIENT CREDENTIALS GRANT

RESOURCE OWNER GRANT

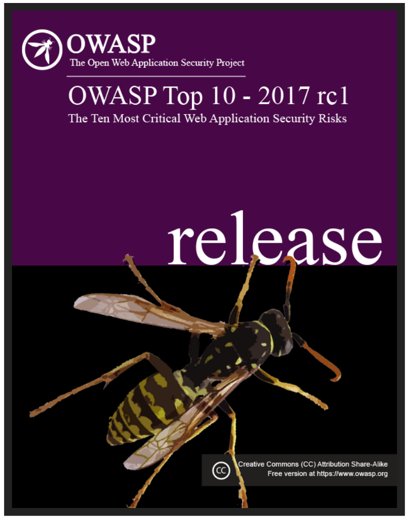
```
POST /token
Host: localhost:9090
Accept: application/json
Content-type: application/x-www-form-encoded
Authorization: Basic b2F1dGgtY2xpZW50LTE6b2F1dGgt...
grant_type=password&scope=openid&username=adm&password=secret
```

DO NOT USE!

WHAT CAN I ACCESS?

A4: BROKEN ACCESS CONTROL

A10: UNDERPROTECTED APIS



OWASP
The Open Web Application Security Project

OWASP Top 10 - 2017 rc1
The Ten Most Critical Web Application Security Risks

release

Creative Commons (CC) Attribution Share-Alike
Free version at https://www.owasp.org

AUTHORIZATION OF REST API

ROLE BASED

```
public class UserBoundaryService {

    @PreAuthorize("hasRole('ADMIN')")
    public List<User> findAllUsers() {...}

}
```

AUTHORIZATION OF REST API

PERMISSION BASED

```
public class TaskBoundaryService {

    @PreAuthorize("hasPermission(#taskId, 'TASK', 'WRITE')")
    public Task findTask(UUID taskId) {...}

}
```

AUTHORIZATION OF REST API

INTEGRATION TEST

```java
public class AuthorizationIntegrationTest {

    @WithMockUser(roles = "ADMIN")
    @Test
    public void verifyFindAllUsersAuthorized() {...}

    @WithMockUser(roles = "USER")
    @Test(expected = AccessDeniedException.class)
    public void verifyFindAllUsersUnauthorized() {...}

}
```

DEMO

WHAT ABOUT THE CLOUD?

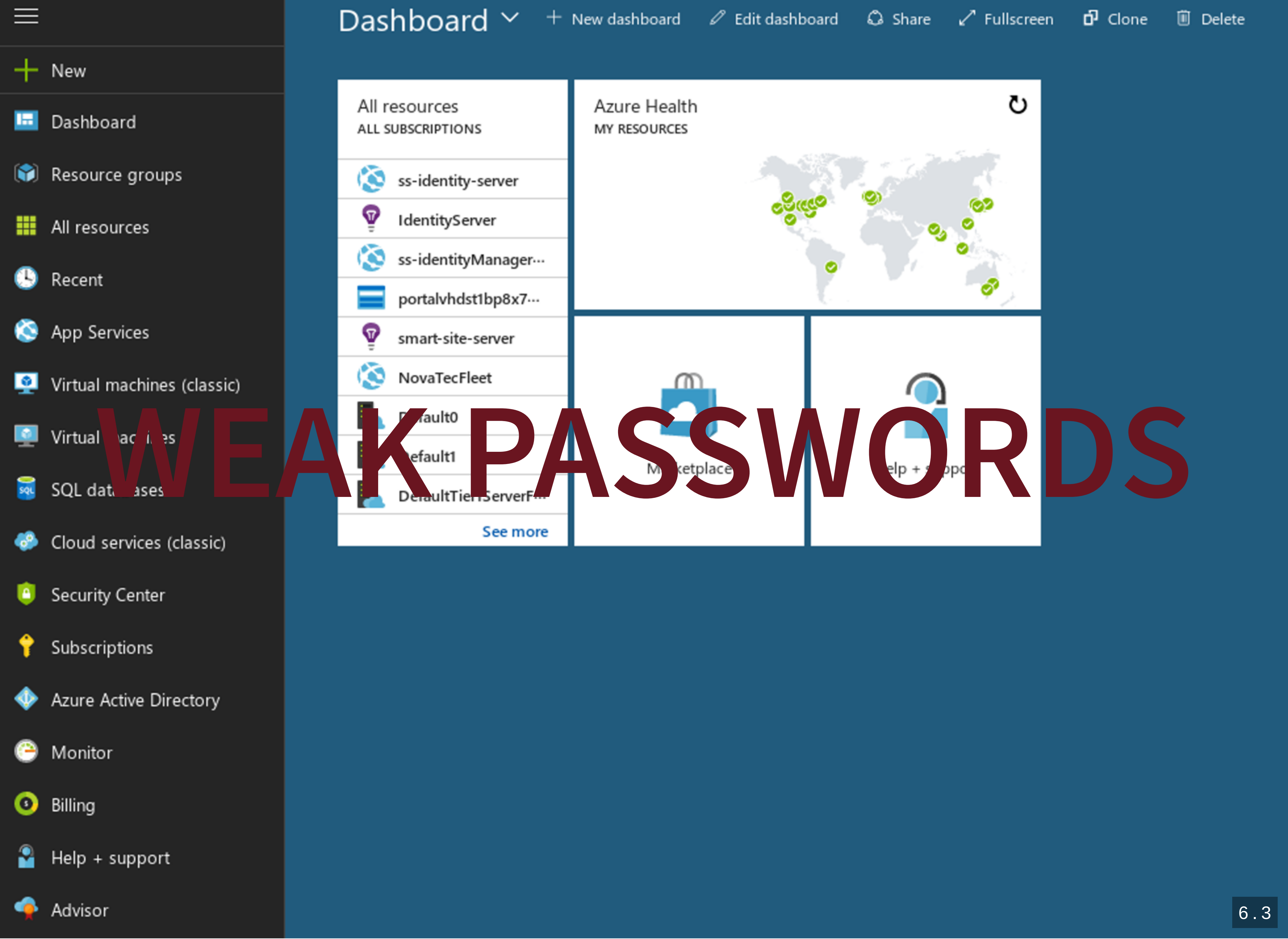
GOOD OLD FRIENDS ...UND MORE...

CSRF XSS SQL Injection Session Fixation Vulnerable Dependencies Weak Passwords Broken Authorization Sensitive Data Exposure

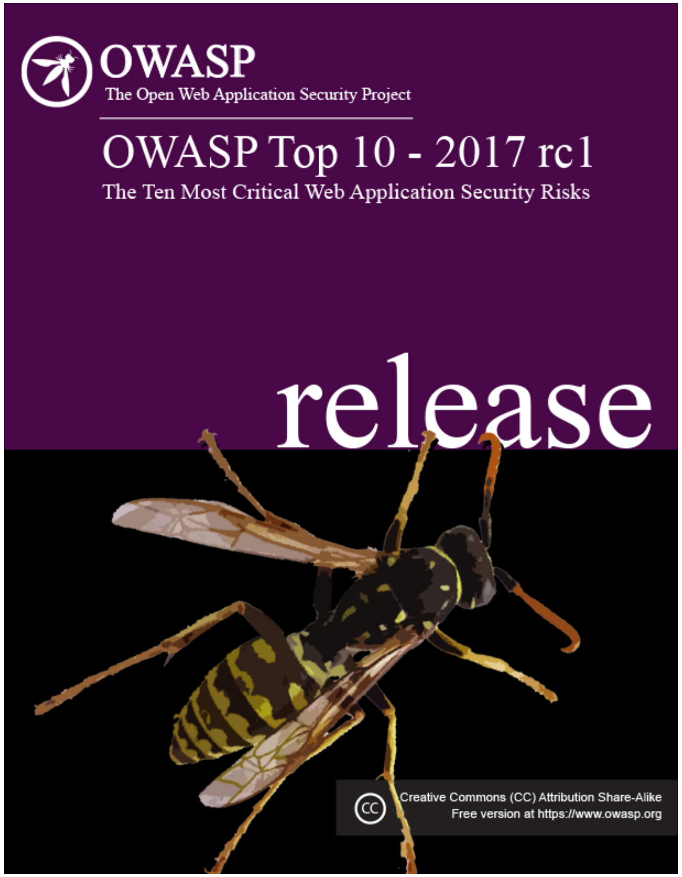
Distributed DoS

Economic DoS

WEAK PASSWORDS



A6: SENSITIVE DATA EXPOSURE



OWASP
The Open Web Application Security Project

OWASP Top 10 - 2017 rc1

The Ten Most Critical Web Application Security Risks

release

Creative Commons (CC) Attribution Share-Alike
Free version at https://www.owasp.org

https://github.com/OWASP/Top10

SPRING CLOUD CONFIG

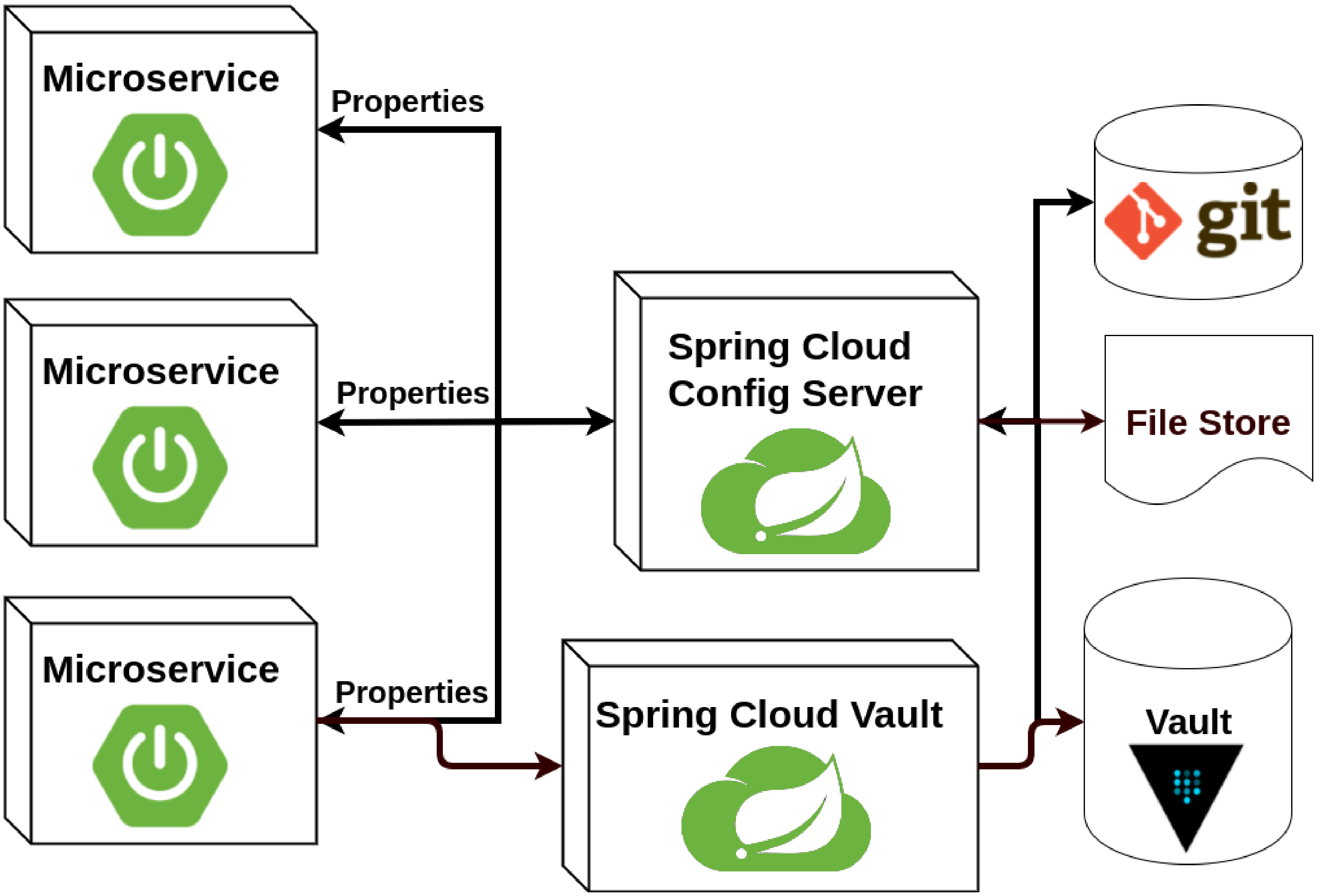
https://cloud.spring.io/spring-cloud-config

Externalized configuration in a distributed system

HTTP, resource-based API

Supports property file and YAML formats

Encrypt and decrypt property values


Microservice
Properties
Microservice
Properties
Microservice
Properties
Spring Cloud Config Server
Spring Cloud Vault
git
File Store
Vault

HashiCorp

Vault

https://www.vaultproject.io/

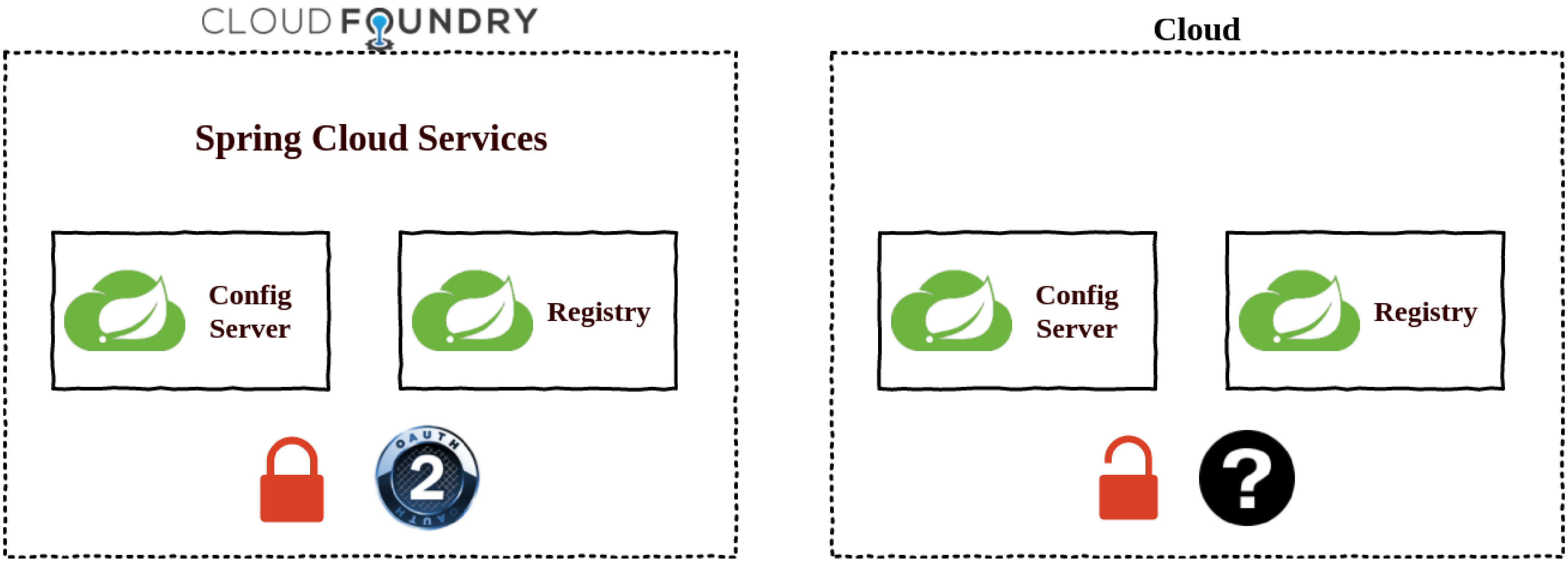
SECRET STORAGE

KEY REVOCATION

KEY ROLLING

AUDIT LOGS

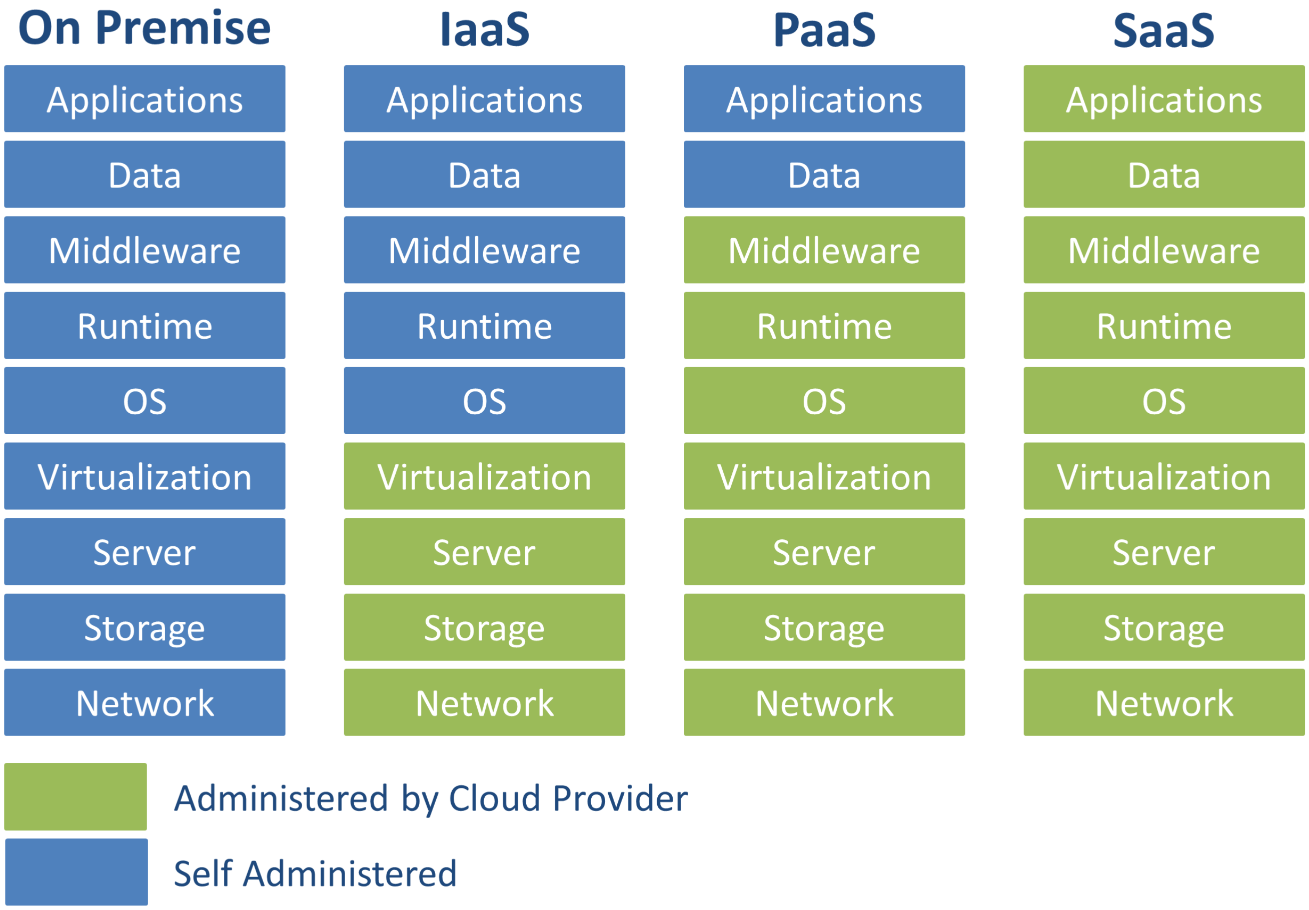
SPRING CLOUD SERVICES SECURITY



SO WHAT IS DIFFERENT

IN THE CLOUD?

Cloud Service Models



On Premise	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟩 Applications
🟦 Data	🟦 Data	🟦 Data	🟩 Data
🟦 Middleware	🟦 Middleware	🟩 Middleware	🟩 Middleware
🟦 Runtime	🟦 Runtime	🟩 Runtime	🟩 Runtime
🟦 OS	🟦 OS	🟩 OS	🟩 OS
🟦 Virtualization	🟩 Virtualization	🟩 Virtualization	🟩 Virtualization
🟦 Server	🟩 Server	🟩 Server	🟩 Server
🟦 Storage	🟩 Storage	🟩 Storage	🟩 Storage
🟦 Network	🟩 Network	🟩 Network	🟩 Network

🟩 Administered by Cloud Provider

🟦 Self Administered

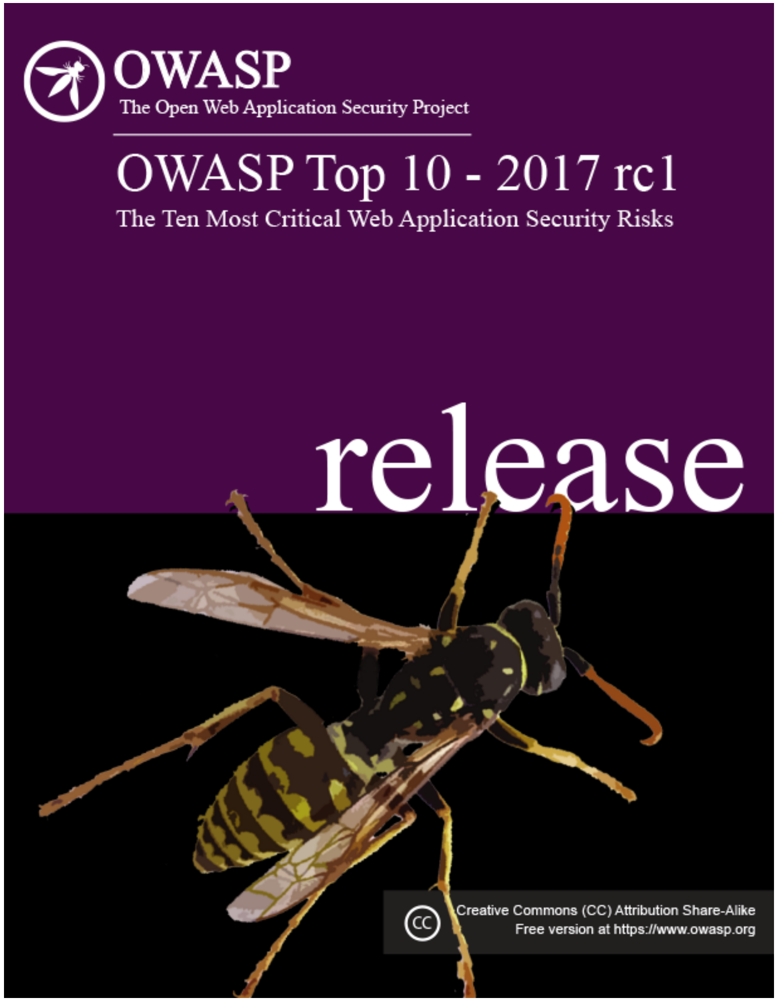
ROTATE, REPAIR, REPAVE

JUSTIN SMITH

“What if every server inside my data center had a maximum lifetime of two hours? This approach would frustrate malware writers…”

ONE MORE THING...

A7: INSUFFICIENT ATTACK PROTECTION



OWASP

The Open Web Application Security Project

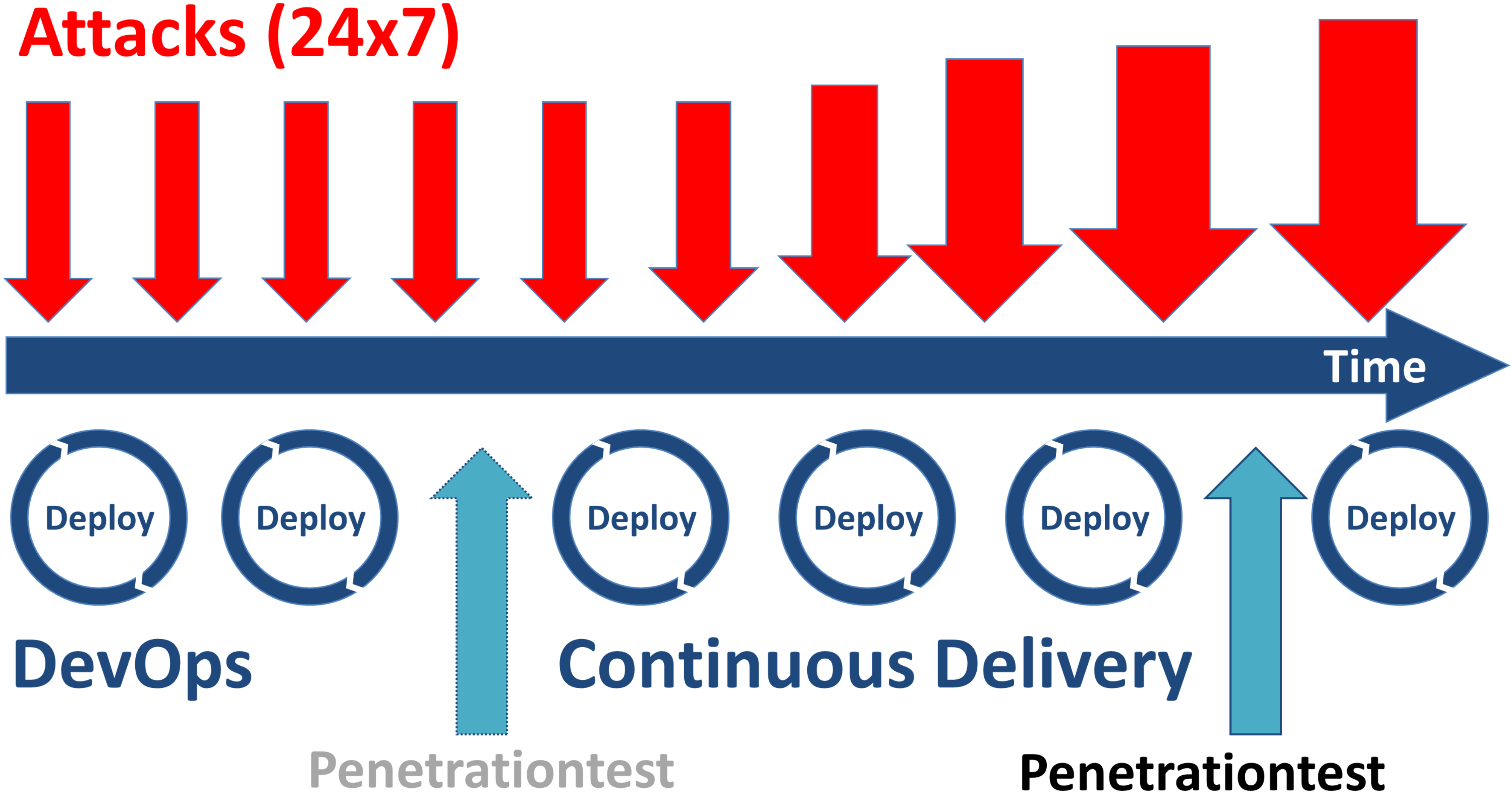
OWASP Top 10 - 2017 rc1

The Ten Most Critical Web Application Security Risks

release

Creative Commons (CC) Attribution Share-Alike

Free version at https://www.owasp.org



Attacks (24x7)

Time

Deploy

Deploy

Deploy

Deploy

Deploy

Deploy

DevOps

Continuous Delivery

Penetrationtest

Penetrationtest

TEST YOUR APPLICATION

BEFORE THE ATTACKER DOES

- OWASP ZAP (https://github.com/zaproxy/zaproxy)
- Burp Suite Free Ed. (https://portswigger.net/burp)
- NMap (https://nmap.org)
- SQLMap (http://sqlmap.org)

REFERENCES

- OWASP Top 10 2017 (https://github.com/OWASP/Top10)
- Application Security Verification Standard (https://github.com/OWASP/ASVS)
- Pro Active Controls (https://www.owasp.org/index.php/OWASP_Proactive_Controls)
- Angular Sandbox Removal (https://angularjs.blogspot.de/2016/09/angular-16-expression-sandbox-removal.html)
- Angular Security Tracking Issue (https://github.com/angular/angular/issues/8511)
- OAuth 2.0 Threat Model and Security Considerations (https://tools.ietf.org/html/rfc6819)
- Implicit Client Implementer's Guide (https://openid.net/specs/openid-connect-implicit-1_0.html)
- Rotate, Repair, Repave (https://thenewstack.io/cloud-foundrys-approach-security-rotate-repair-repave)
- Spring Cloud Config (https://cloud.spring.io/spring-cloud-config/)
- Spring Cloud Vault (https://cloud.spring.io/spring-cloud-vault)
- Vault (https://www.vaultproject.io)

Q&A

http://www.novatec-gmbh.de
http://blog.novatec-gmbh.de

andreas.falk@novatec-gmbh.de

@NT_AQE, @andifalk